Calculating Velocities of Cloud Features Across a Great Circle Arc in Venus's Atmosphere

In the discussion below we derive two different methods for calculating the velocities of cloud silhouettes across a great circle arc in Venus's atmosphere. The cloud silhouettes are created by heat in the deep atmosphere of Venus, and are observed using infrared images taken by NASA's Infrared Telescope Facility (IRTF) in Hawai'i. The velocity of the cloud silhouettes give insight into the wind speeds at the altitude they are observed. The two methods for calculating velocity assume the shortest distance of travel is taken by the cloud silhouettes. The first method utilizes the haversine formula first published by R. W. Sinnott, "Virtues of the Haversine", *Sky and Telescope* **68** (2), 159 (1984). This method is used frequently on earth in aviation. The second method is derived using angular velocity. The second method was looked at as a method to check the accuracy of the first method.

1.1 Haversine Formula:

The trigonometric function, haversine, is defined as:

$$\mathrm{haversin}(\theta) = \sin^2\left(\frac{\theta}{2}\right) \tag{1}$$

For two points on a sphere of radius R, the distance d, between two points along a great circle arc of the sphere is related to their locations by the formula:

$$\mathrm{haversin}\left(\frac{\mathrm{d}}{\mathrm{R}}\right) = \mathrm{haversin}(\Delta\beta) + \cos(\beta_1)\cos(\beta_2)\mathrm{haversin}(\Delta\alpha) \tag{2}$$

Where $\Delta\alpha = \alpha_1 - \alpha_2$ is the longitude separation and $\Delta\beta = \beta_1 - \beta_2$ is the latitude separation and angles are measured in radians.

One can then solve for d by simply applying the inverse haversine function:

$$d = R\,\mathrm{haversine}^{-1}(\mathrm{haversin}(\Delta\beta) + \cos(\beta_1)\cos(\beta_2)\mathrm{haversin}(\Delta\alpha)) \tag{3}$$

We note here that $d = R\text{haversine}^{-1}(\text{haversin}(\Delta\beta))$ represents the latitudinal component of the distance and $d = R\text{haversine}^{-1}(\cos(\beta_1)\cos(\beta_2)\text{haversin}(\Delta\alpha))$ represents the longitudinal component of the distance.

For accessibility purposes we chose Microsoft Excel as our medium for carrying out our calculations. Since Microsoft Excel does not recognize the haversine function, we can simply convert all haversine's to sine and inverse sine functions using definition (1):

$$d = 2R\sin^{-1}\left(\sqrt{\sin^2\left(\frac{\Delta\beta}{2}\right) + \cos(\beta_1)\cos(\beta_2)\sin^2\left(\frac{\Delta\alpha}{2}\right)}\right) \tag{4}$$

1.2 Canonical Cartesian to Spherical Coordinate Transformations:

The canonical Cartesian to spherical coordinate transformations are given as*:

$$r = \sqrt{x^2 + y^2 + z^2} \tag{5}$$

$$\theta = \tan^{-1}\left(\frac{y}{x}\right) \tag{6}$$

$$\varphi = \tan^{-1}\left(\frac{\sqrt{x^2+y^2}}{z}\right) \tag{7}$$

Where r is the radius, θ is measured in radians circularly about the z axis starting from the positive x axis, and φ is measured in radians circularly about the origin starting from the positive z axis. This means that $\theta \cong \alpha$ and we will set the range of both to be [90 → -90]. On the other hand β and φ are going to be definitionally related by $\beta = 90 - \varphi$ in the northern hemisphere and $\beta = -90 - \varphi$ in the southern hemisphere. We can set the range to be [90 → -90] for both of these as well.

*Since our actual measurement of distance is in number of pixels, all of the canonical coordinates are to be defined in pixel number which will then be convert to mks units. Hence the reason for two different symbols for radius.
For example: r [pixels] → R [mks]

1.3 Using the 2D Data to Calculate the True Distance and Velocity :

Beginning with our 2D image of Venus we note that pixel (0, 0), the origin in the computers mind, starts in the upper left corner of the image. All pixels have a positive value. x increasing to the right and y increasing downward. In order to create a set of spherical coordinates around our 2D image of Venus we must first determine which (x, y) pixel values correspond to Venus's center and shift that pixel to the center of the image (256, 256). This will be our new origin. To keep our coordinates right handed we will now use (y, z) so that positive y will point to the right, positive z will point up, and positive x will point out of the page. This step will help us keep everything in a standard frame of reference. It is also necessary because spherical coordinates require negative (x, y, z) values to complete all 6 sextants of the sphere.

We next need to determine the radius of Venus in number of pixels. This is because spherical coordinates rely on a radius for positioning. This will also allow us to determine our distance scale, $[\frac{km}{pixel}]$.

Once we have set the center and found the radius we will have enough information to use the canonical coordinate transformations. The knowns in the canonical equations are: z, y, and r. The unknowns are: x, θ, φ.

The two important pieces of information we wish to extract are θ and φ because those values directly relate to longitude and latitude respectively. To find these values we must massage them until they are independent of x. This is easy since we have 3 equations and 3 unknowns. Solving for x is simple because we presumably know r:

$$x = \sqrt{r^2 - z^2 - y^2} \tag{8}$$

Now we have a θ and φ independent of x:

$$\theta = \tan^{-1}\left(\frac{y}{\sqrt{r^2 - z^2 - y^2}}\right) \tag{9}$$

$$\varphi = \tan^{-1}\left(\frac{\sqrt{r^2 - z^2}}{z}\right) \tag{10}$$

Once we obtain values for θ and φ for two independent points, we can relate them to α and β to extract the distance and the wind speed from the haversine formula or the angular velocity formula.

The accompanying Microsoft Excel worksheet is designed to automatically perform all of these calculations based on only 6 inputs from the user: y_1, z_1, y_2, z_2, r and Δt, the elapsed time between image 1 and image 2 in hours. All input variables are marked by big red boxes.

2.1 Angular Velocity:

The dot product of two vectors,

$$\vec{P}_1 = (x_1, y_1, z_1)$$

$$\vec{P}_2 = (x_2, y_2, z_2)$$

Is defined as:

$$\vec{P}_1 \cdot \vec{P}_2 = |P_1||P_2| \cos\theta \tag{11}$$

$$= x_1 x_2 + y_1 y_2 + z_1 z_2 \tag{12}$$

Where θ is the angle between $\vec{P}_1$ and $\vec{P}_2$ and $|P_1| = |P_2| = R$.

We can solve for θ:

$$\theta = \cos^{-1}\left(\frac{\vec{P}_1 \cdot \vec{P}_2}{|P_1||P_2|}\right) \tag{13}$$

$$\boxed{= \cos^{-1}\left(\frac{x_1 x_2 + y_1 y_2 + z_1 z_2}{R^2}\right)} \tag{14}$$

Angular velocity is a pseudovector quantity which specifies the angular speed of an object and the axis about which the object is rotating. The SI unit is radians per second. The rate of change of the angular position of a particle is related to the cross-radial velocity by:

$$v = R\frac{d\theta}{dt} \tag{15}$$

Where R, the radius, is defined using the canonical spherical coordinate transformation* $R = \sqrt{x^2 + y^2 + z^2}$ and $\frac{d\theta}{dt}$ is simply θ (the angle sweeping the difference in position between $\overrightarrow{P_1}$ and $\vec{P}_2$) divided by the time interval:

$$\frac{d\theta}{dt} = \frac{\theta}{t_2 - t_1} \tag{16}$$

Therefore the angular velocity is given as:

$$v = R\frac{\theta}{t_2 - t_1} \tag{17}$$

$$\boxed{= R\frac{\cos^{-1}(\frac{x_1x_2 + y_1y_2 + z_1z_2}{R^2})}{t_2 - t_1}} \tag{18}$$

We note here that equation (18) allows us to directly calculate the wind speed over the great circle arc. It is independent of latitude and longitude. In fact the only measurements we need for this is method is the radius and the (x, y, z) coordinates of two points.

The total distance traveled is given by:

$$d_{tot} = R\cos^{-1}(\frac{x_1x_2 + y_1y_2 + z_1z_2}{R^2}) \tag{19}$$

*In the Excel sheet we use simple relationships to calculate all distances in true distance and in pixel number. The angular velocity formula will agree with both distance scales.

3.1 Conclusion:

After taking our measurements and inputting the data we find that both the haversine method and the angular velocity method agree to a surprising degree of accuracy. It is reasonable to conclude that this is not a coincidence. These measurements return wind speeds on the order in which we would expect. It is known that the haversine formula is more accurate on very small distance scales (on the order of centimeters) due to error in floating point calculations in the inverse cosine function. However our measurements are not that precise. It should be noted that at greater distance/time intervals these measurements return more accurate speeds. On scales of 4 - 5 pixels the error in our measurements seems to dominate.